**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра Вычислительной техники**

отчет

**по лабораторной работе № 4**

**по дисциплине «Программирование»**

Тема: Применение функций

|  |  |  |
| --- | --- | --- |
| Студент гр. 3312 |  | Шарапов И. Д. |
| Преподаватель |  | Аббас С. А. |

Санкт-Петербург

2023

**Содержание**

[Цель работы 3](#_Toc150691142)

[Задание (Вариант 10) 3](#_Toc150691143)

[Постановка задачи и описание решения 3](#_Toc150691144)

[Описание переменных 4](#_Toc150691145)

[Схема алгоритма 5](#_Toc150691146)

[Текст программы 6](#_Toc150691147)

[Контрольные примеры 7](#_Toc150691148)

[Примеры выполнения программы 7](#_Toc150691149)

[Выводы 8](#_Toc150691150)

# Цель работы

Целью работы является изучение особенностей работы с функциями в языке Си и получение практических навыков в решение задач, в которых необходимо обрабатывать значения матрицы.

# Задание (Вариант 10)

Ввести построчно элементы двумерного массива чисел заданных размеров. Вывести исходный массив. Из строк исходного массива, в которых содержится заданное после ввода массива количество одинаковых элементов, сформировать столбцы результирующего массива. Вывести сформированный массив. Вывод строки массива и подсчёт количества одинаковых элементов оформить в виде функций.

# Постановка задачи и описание решения

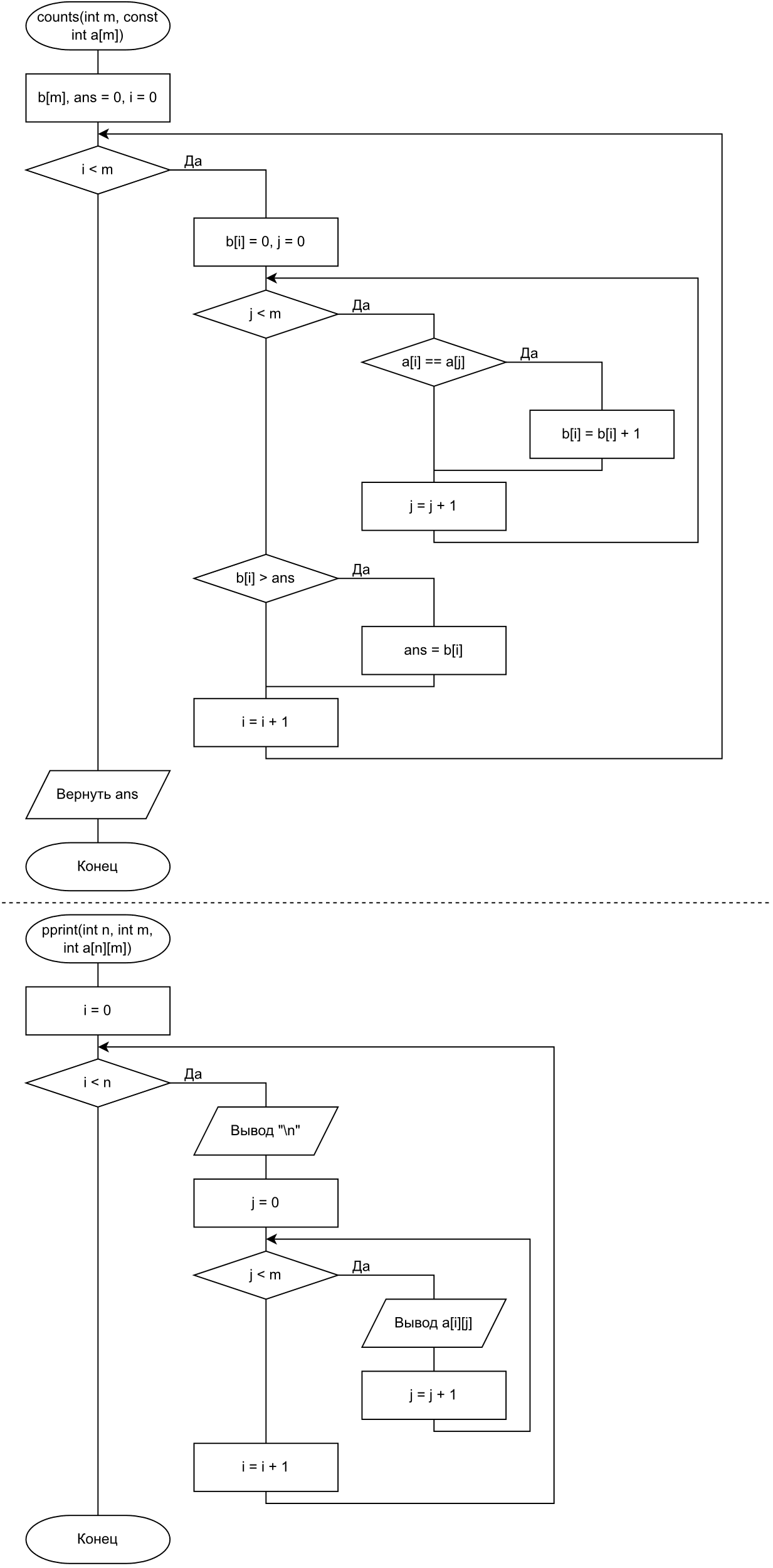
Для решения задачи используется 2 функции. Функция *pprint(int n, int m, int a[n][m])* используется для вывода двумерного массива. С помощью вложенных циклов перебираются и выводятся все элементы массива. Функция *counts(int m, const int a[m])* используется для того, чтобы посчитать какое максимальное количество раз повторяется какой-то из элементов этого массива. Данный алгоритм основывается на полном переборе: каждый элемент массива мы сравниваем с каждым элементом этого же массива. В случае совпадения i-ого и j-ого элементов прибавляем единичку в вспомогательный массив b (b[i] += 1). В процессе ищем максимальное значение в массиве b и храним его в переменной ans. В конце возвращаем ans.

В целом программа работает следующим образом. Сначала считываем размеры матрицы. Затем построчно считываем элементы матрицы. Спрашиваем у пользователя количество повторений. Выводим матрицу с помощью функции *pprint*. Далее перебираем все строки матрицы и считаем количество строк, в которых повторений больше, либо равно заданному, используя функции *counts*. Если строк, удовлетворяющих условию нет: выводим *“The matrix is empty!”.* Иначе инициализируем массив res[m][cnt], где cnt количество строк, удовлетворяющих условию. Далее все строки матрицы a, удовлетворяющие условию, с помощью цикла записываем в соответствующие столбцы матрицы res. Выводим сообщение *"Required matrix:".* И с помощью функции pprint выводим массив res.

# Описание переменных

|  |  |  |  |
| --- | --- | --- | --- |
| Функция *pprint(int n, int m, int a[n][m])* | | | |
| № | Имя переменной | Тип | Назначение |
| 1 | n | int | Количество строк |
| 2 | m | int | Количество столбцов |
| 3 | a[n][m] | int[][] | Массив, который нужно вывести |
| Функция *counts(int m, const int a[m])* | | | |
| № | Имя переменной | Тип | Назначение |
| 1 | m | int | Количество элементов в строке |
| 2 | a[m] | int[] | Строка, в которой нужно найти наибольшее повторение элементов |

# Схема алгоритма



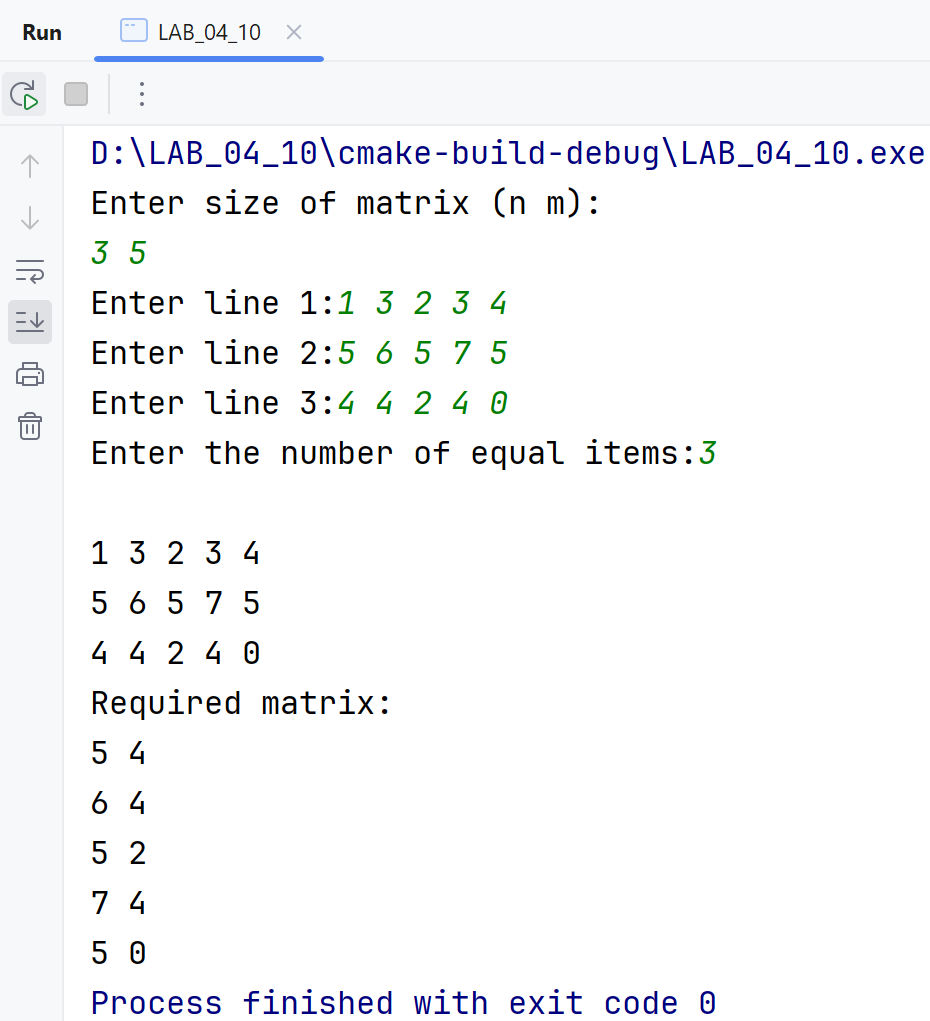
# Текст программы

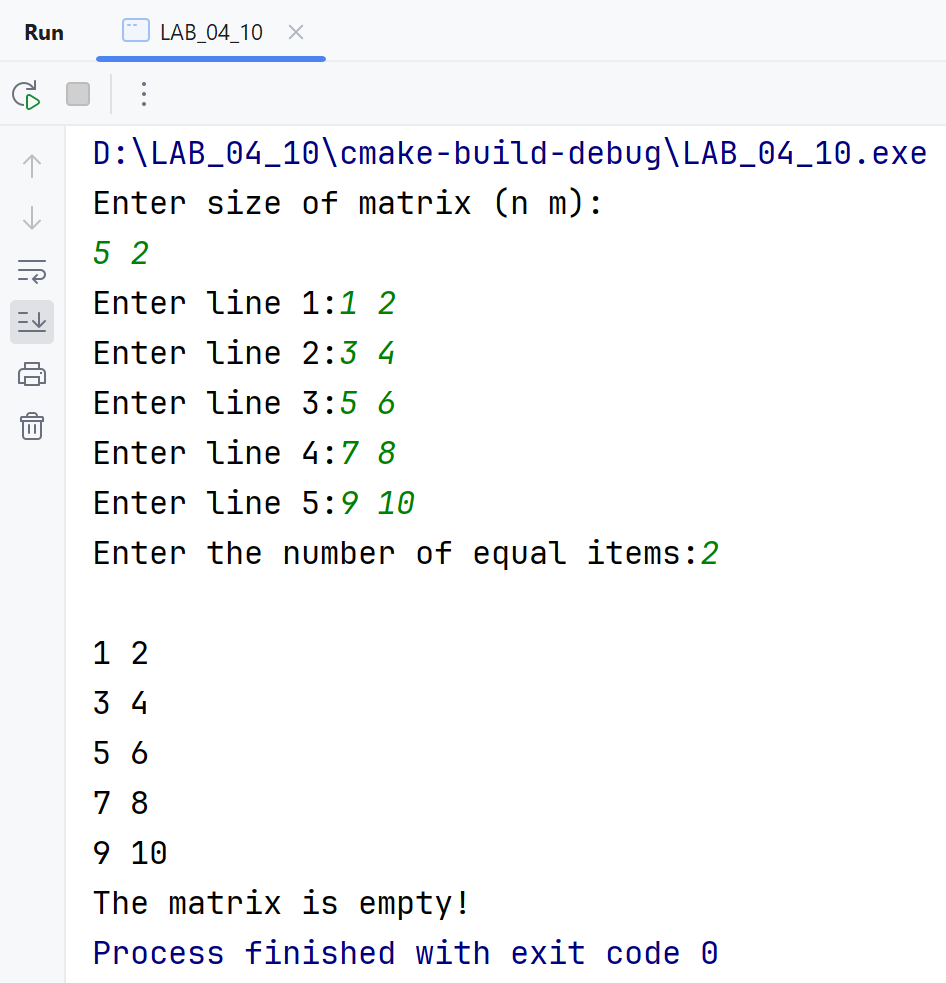
|  |
| --- |
| #include <stdio.h>  void pprint(int n, int m, int a[n][m]) {  for (int i = 0; i < n; ++i) {  printf("\n");  for (int j = 0; j < m; ++j) {  printf("%i ", a[i][j]);  }  } }  int counts(int m, const int a[m]) {  int b[m], ans = 0;  for (int i = 0; i < m; ++i) {  b[i] = 0;  for (int j = 0; j < m; ++j) {  if (a[i] == a[j]) {  ++b[i];  }  }  if (b[i] > ans) {  ans = b[i];  }  }  return ans; }  int main() {  printf("Enter size of matrix (n m):\n");  int n, m, k, cnt, ind;  scanf("%i %i", &n, &m);  int a[n][m];  for (int i = 0; i < n; ++i) {  printf("Enter line %i:", i + 1);  for (int j = 0; j < m; ++j) {  scanf("%i", &a[i][j]);  }  }   printf("Enter the number of equal items:");  scanf("%i", &k);  pprint(n, m, a);   cnt = 0;  for (int i = 0; i < n; ++i) {  if (counts(m, a[i]) >= k) {  ++cnt;  }  }  if (cnt == 0) {  printf("\nThe matrix is empty!");  } else {  ind = 0;  int res[m][cnt];  for (int i = 0; i < n; ++i) {  if (counts(m, a[i]) >= k) {  for (int j = 0; j < m; ++j) {  res[j][ind] = a[i][j];  }  ++ind;  }  }  printf("\nRequired matrix:");  pprint(m, cnt, res);  }  return 0; } |

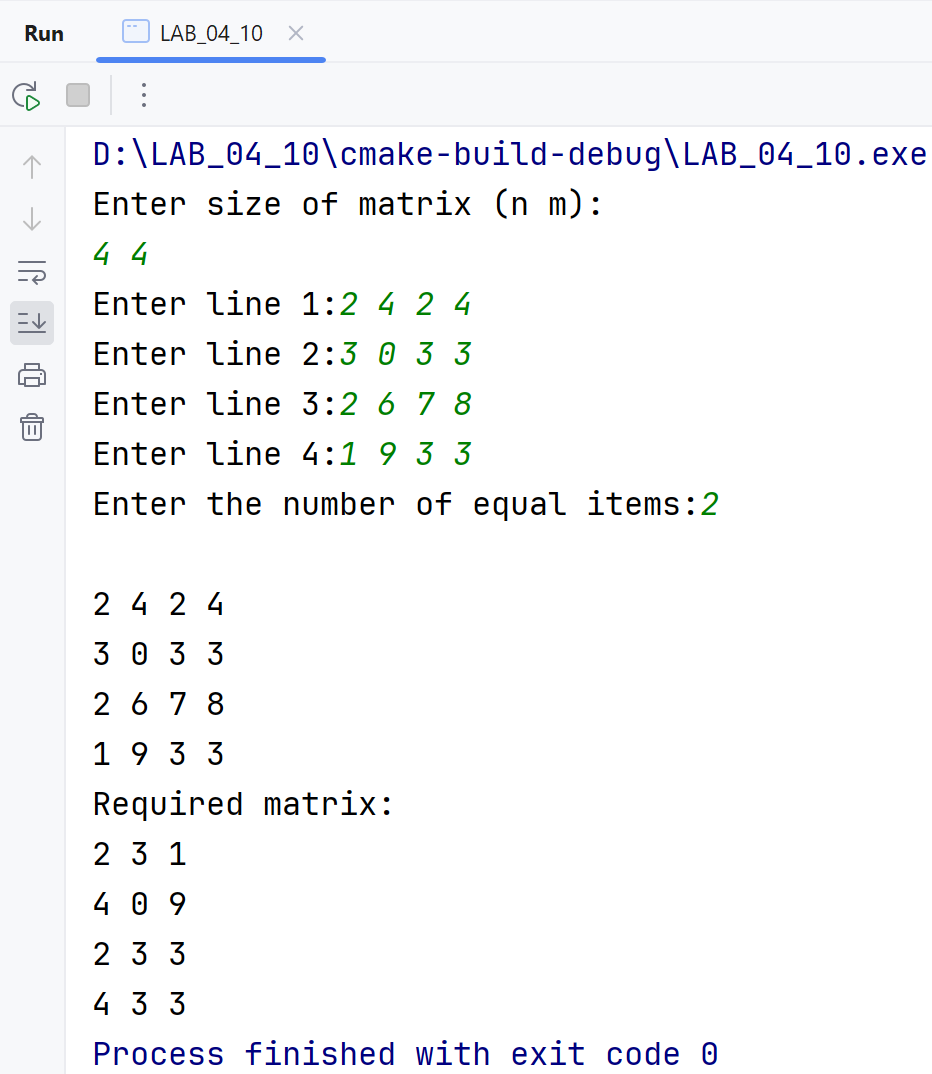
# Контрольные примеры

|  |  |  |
| --- | --- | --- |
| № | Исходные данные | Результаты |
| 1 | 3 5  1 3 2 3 4  5 6 5 7 5  4 4 2 4 0  3 | 1 3 2 3 4  5 6 5 7 5  4 4 2 4 0  Required matrix:  5 4  6 4  5 2  7 4  5 0 |
| 2 | 5 2  1 2  3 4  5 6  7 8  9 10  2 | 1 2  3 4  5 6  7 8  9 10  The matrix is empty! |
| 3 | 4 4  2 4 2 4  3 0 3 3  2 6 7 8  1 9 3 3  2 | 2 4 2 4  3 0 3 3  2 6 7 8  1 9 3 3  Required matrix:  2 3 1  4 0 9  2 3 3  4 3 3 |

# Примеры выполнения программы







# Выводы

В результате выполнения работы изучены особенности функций в языке Си. А также получены практические навыки в решении задач, связанных с матрицами.